Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 4 (2011) 1422–1430

Procedia Computer Science

A Trust-Region Algorithm for Bi-Objective Stochastic Optimization

Sujin Kim[a,1], Jong-hyun Ryu[a,*]

[a]*National University of Singapore 1 Engineering Drive 2, 117576, Singapore*

Abstract

We develop a new method for approximating the Pareto front of a bi-objective stochastic optimization problem in which the expected objective functions are estimated by taking sample averaged outputs from expensive simulations. At each iteration of the proposed algorithm, a trust region is identified and quadratic approximate functions for the expected objective functions are built using the sample average values. To determine non-dominated solutions in the trust region, a single-objective optimization problem is constructed based on the approximate objective functions. After updating the set of non-dominated solutions, a new trust region around the most isolated point is determined to explore areas that have not been visited. When the computational budget is limited, a large sample size at each iteration leads to more accurate approximation of the expected objective functions, but the algorithm is not able to run for enough iterations to generate a set of solutions that are close to the Pareto front. The proposed variable sampling scheme adaptively updates the sample size with consideration for this trade-off between approximation and optimization errors. The numerical results show that our proposed method is feasible, and the performance can be significantly improved with an appropriate sampling scheme.

Keywords: multi-objective optimization, stochastic programming, sample average approximation, trust-region method,

1. Introduction

Multi-objective optimization problems arise in a wide range of areas such as economics and engineering. For example, a product designer might want to simultaneously optimize several critical factors [1, 2, 3], or a financial portfolio manager looks for a position where both return and risk can be optimized [4]. Different objectives often conflict with each other, and a single decision cannot optimize all the objectives at the same time. Therefore, the notion of optimality in single-objective optimization cannot be directly applied to multi-objective optimization problems, and it is necessary to introduce a new concept of optimality in this framework. The optimal decision may be taken by identifying the best trade-offs among the multiple objectives, which is often the goal of multi-objective optimization. More specifically, the best trade-off solutions can be described by the notion of Pareto optimality or Pareto efficiency, which has been regarded as an important criterion of performance for multi-objective optimization in economics and engineering systems.

*Research fellow in the department of Industrial Systems & Engineering

Email addresses: isesk@nus.edu.sg (Sujin Kim), iserj@nus.edu.sg (Jong-hyun Ryu)

[1]Assistant professor in the department of Industrial Systems & Engineering

Selection and/or peer-review under responsibility of Prof. Mitsuhisa Sato and Prof. Satoshi Matsuoka

doi:10.1016/j.procs.2011.04.153

When the number of objectives is p, the multi-objective optimization problem can be formally expressed as follows:

$$\min_{x \in \mathbf{X}} F(x) = (f_1(x), f_2(x), ..., f_p(x)),$$

where $\mathbf{X}$ is a subset of $\mathbb{R}^q$ and $F : \mathbf{X} \to \mathbb{R}^p$ is a vector valued function. In this paper, we assume that $\mathbf{X}$ is continuous. Pareto optimality for the problem with the above form is defined by the dominance relationship [5].

Definition 1. *Let x_u, $x_v \in \mathbf{X}$ be two decision vectors. $F(x_u)$ dominates $F(x_v)$ (denoted by $F(x_u) \prec F(x_v)$) if and only if $f_i(x_u) \leq f_i(x_v)$, for all $i \in \{1, 2, ..., p\}$ and there exists $l \in \{1, 2, ..., p\}$ such that $f_l(x_u) < f_l(x_v)$.*

Definition 2. *A point $x^* \in \mathbf{X}$ is said to be globally Pareto optimal if and only if there is no $x \in \mathbf{X}$ such that $F(x) \prec F(x^*)$. $F(x^*)$ is called globally efficient and the image of the set of globally efficient points is called the Pareto front.*

In the case where the Pareto front is disconnected, it is hard to develop a method that can guarantee the global Pareto optimality for a general class of problems. In this work, we focus on searching for a Pareto front in a local region instead.

Definition 3. *A point $x^* \in \mathbf{X}$ is said to be locally Pareto optimal if and only if there exists an open neighborhood of x^*, $\mathfrak{N}(x^*)$, such that there is no $x \in \mathfrak{N}(x^*) \cap \mathbf{X}$ satisfying $F(x) \prec F(x^*)$. $F(x^*)$ is then called locally efficient and the image of the set of locally efficient points is called the local Pareto front.*

One of the most widely used approaches to solve multi-objective optimization problems in the above form is to convert them into single-objective problems. The weighted sum method (WS) [3, 6] takes a convex combination of the multiple objectives and minimizes it. By taking different weight combinations, a desired number of optimal solutions can be identified and used to approximate the Pareto front. However, the WS method has a serious limitation: it is able to find points only on the convex part of the Pareto front. Audet et al. [7] focuses particularly on a class of bi-objective problems and applies a direct search method to a series of single-objective problems that are designed to detect solutions on both convex and nonconvex parts of Pareto front. The algorithm generates a set of points converging to the Pareto front, but the convergence rate may be slow when the dimension of the considered problem is high. Ryu et al. [8] developed a trust-region method for the bi-objective black-box optimization problem. Their numerical results show that the proposed algorithm performs well for both convex and nonconvex problems and outperforms the method developed by Audet et al. [7] for a certain set of problems. For large-scale, complex problems, an evolutionary algorithm combined with the weighted sum method can be considered [5, 9]. The algorithm is not guaranteed to converge, and can be computationally expensive because a massive number of non-Pareto set points should be evaluated while searching for Pareto optimal solutions.

In this paper, we consider black-box problems where the analytic forms of the objective functions $f_i, i = 1, \ldots, p$, are not available, and the values can only be estimated by output responses from computationally expensive simulations. In addition, we assume that the estimated function values contain stochastic white noise. We are particularly interested in unconstrained stochastic bi-objective problems with the following form:

$$\min_{x \in \mathbb{R}^q} \{F(x) = (f_1(x), f_2(x)) := EF(x, \xi) = E\left[f_1(x, \xi_1), f_2(x, \xi_2)\right]\}. \tag{1}$$

ξ_1 and ξ_2 are random vectors defined on a probability space $(\Omega, \mathcal{F}, P)$ with support Ξ_1 and Ξ_2, respectively. The sample objective functions $f_i(\cdot, \cdot) : \mathbb{R}^q \times \Xi_i \to \mathbb{R}$, $i = 1, 2$, are real valued functions, and their values are evaluated via simulation. The value of the expected functions cannot be computed exactly, but can be estimated via sample averages. We assume that the two objective functions f_1 and f_2 are in conflict with each other and have finite minimum values.

In general, multi-objective optimization is considered to be difficult because the goal is to find a set of solutions that can accurately approximate the Pareto front, rather than search for a single solution. The stochastic noise in the sample response adds another layer of complexity and challenge to the problem. A very limited work is found in the literature in this area. Lee et al. [10] develop a ranking & selection procedure for a multi-objective problem with a finite number of feasible designs.

To approximate the problem (1), we take the sample average approximation (SAA) approach. The basic idea of SAA is to replace the expected value function by the sample average, and then solve the resulting problem using a deterministic optimization algorithm [11]. The SAA problem to the problem (1) is defined by

$$\min_{x\in\mathbb{R}^q}\left\{\hat{F}(x)=\left(\hat{f}_1(x),\hat{f}_2(x)\right)=\left(\frac{1}{N}\sum_{r=1}^{N}f_1(x,\xi_{1r}),\frac{1}{N}\sum_{r=1}^{N}f_2(x,\xi_{2r})\right)\right\},\tag{2}$$

where N is the sample size and $(\xi_{11},\xi_{21}),(\xi_{12},\xi_{12}),\ldots,(\xi_{1N},\xi_{1N})$ are N independent replications of (ξ_1,ξ_2).

To identify non-dominated solutions, we modify the trust-region method for multi-objective deterministic optimization developed by Ryu et al. [8] and apply it to the SAA problem (2). At each iteration of the proposed algorithm, a trust region is identified and quadratic approximate functions for the expected objective functions are built using the sample average values at design points selected within a trust region. A single-objective optimization problem is constructed based on the approximate objective functions to search for non-dominated solutions in the trust region. After updating the set of non-dominated solutions, a new trust region around the most isolated point is determined to explore the non-visited area.

The accuracy of solutions from the algorithm can be improved by controlling sampling and optimization errors. The difference between the solutions obtained from SAA and the solutions to the true problems can be reduced by taking a larger sample size at each iteration. On the other hand, as the number of iterations grows, the distance between solutions from each iteration and the Pareto front decreases. Also, a bigger number of solutions over a wide region can be evaluated, which allows the method to generate more widely spread and evenly distributed Pareto optimal solutions. When the computational budget is limited, the sample size at each iteration should be carefully chosen with consideration for this trade-off between sampling and optimization errors. We adopt the idea of the variable sampling scheme [12, 13, 14] to adaptively update the sample size based on the result from the past iteration.

This paper is organized as follows. In section 2, we describe the proposed algorithm for solving bi-objective optimization problems in detail. In Section 3, to show the feasibility of our proposed method, we conduct numerical experiments with different sampling schemes. Section 4 draws some concluding remarks.

2. Bi-objective Stochastic Optimization Algorithm

In this section, we consider the bi-objective stochastic optimization problem (1) and present an algorithm to generate a set of Pareto optimal solutions. The algorithm iteratively applies a trust-region method [8, 15] to SAA problems and finds a set of non-dominated points within a local region. At each iteration, the most isolated point is selected among the points that have thus far been determined to be non-dominated. This point is then defined as the current iterate. A trust region centered at the current iterate is determined to maintain the uniformity of the optimal solution set by exploring non-visited areas. Thus, the trust region iteratively moves according to the selected iterate. Several design points in the trust region are chosen using a design of experiment technique, and the sample average values at the design points as well as the iterate are computed. A quadratic regression model for each objective function is constructed based on the sample average values at the design points [16], and a single-objective optimization problem is built to search for non-dominated solutions within the trust region. To improve the efficiency of the algorithm, the sample size at each iteration can be adaptively determined considering the present state of the algorithm such as the number of iteration, the trust region radius and the variance of the sample response. We assume that the objective functions f_1 and f_2 are Lipschitz continuous so that trust region method for single objective problems can converge.

2.1. Iterate determination

We define

$$\mathcal{X}^{(k)}=\{x_j^{(k)},\ j=1,2,\ldots,J^{(k)}\}$$

as a set of non-dominated points returned at the end of iteration $(k-1)$, where $J^{(k)}$ denotes the cardinality of $\mathcal{X}^{(k)}$. The non-dominated point $x_j^{(k)}$ is associated with the following four parameters:

- $n_j^{(k)}$: the sample size used to evaluate the value of sample average objectives at $x_j^{(k)}$.
- $\hat{F}_j^{(k)}=\left(\hat{f}_{1j}^{(k)},\ \hat{f}_{2j}^{(k)}\right)$: the vector of sample average objectives evaluated at $x_j^{(k)}$. For $i=1,2$, $\hat{f}_{ij}^{(k)}=\frac{1}{n_j^{(k)}}\sum_{r=1}^{n_j^{(k)}}f(x_j^{(k)},\xi_{ir})$

- $\Delta_j^{(k)}$ and $\rho_j^{(k)}$: Suppose that $x_j^{(k)}$ is detected at some iteration $l < k$. If Δ is the radius of the trust-region used to construct quadratic approximate models at the iteration l and ρ is the reduction ratio (see Equation (5)) computed at $x_j^{(k)}$, we set $\Delta_j^{(k)} = \Delta$ and $\rho_j^{(k)} = \rho$.

While we would like to identify solutions close to the Pareto front, we also want to generate well-spread solutions in order to approximate as much of the Pareto front as possible. To this end, we select the most isolated point $x_c^{(k)}$ in $\mathcal{X}^{(k)}$ and search for new solutions around a region centered at the point. We introduce a quantity $\gamma_j^{(k)}$ that indicates the degree of isolation of point $x_i^{(k)}$, based on the distance between the objective vectors at $x_j^{(k)}$ and its neighboring points. When measuring the distance between two objective vectors, the sampling error should be taken into consideration. We compute the confidence interval of each objective value and consider the widest possible distance between the two objective vectors. Let $[\underline{C}_{ij}^{(k)}, \overline{C}_{ij}^{(k)}]$ $(i = 1, 2, j = 1, 2, \ldots, J^{(k)})$ be the confidence interval of $\hat{f}_{ij}^{(k)}$ under normality assumptions with a significance level of $1 - \alpha$. The procedure to compute the isolation measure $\gamma_j^{(k)}$, $j = 1, 2, \ldots, J^{(k)}$ and to select the most isolated point $x_c^{(k)}$ is as follows:

1. Sort points in $\mathcal{X}^{(k)}$ so that $\hat{f}_{11}^{(k)} \leq \hat{f}_{12}^{(k)} \leq \ldots \hat{f}_{1J^{(k)}}^{(k)}$.
2. Let $\delta > 0$ be a user-defined constant. We compute $\gamma_j^{(k)}$ using the below formula:

$$\gamma_j^{(k)} = \begin{cases} \max\left(\delta,\ 2\sqrt{\left(\underline{C}_{1j}^{(k)} - \overline{C}_{1(j+1)}^{(k)}\right)^2 + \left(\overline{C}_{2j}^{(k)} - \underline{C}_{2(j+1)}^{(k)}\right)^2}\right) & \text{if } j = 1 \\ \max\left(\delta,\ 2\sqrt{\left(\underline{C}_{1(j-1)}^{(k)} - \overline{C}_{1j}^{(k)}\right)^2 + \left(\overline{C}_{2(j-1)}^{(k)} - \underline{C}_{2j}^{(k)}\right)^2}\right) & \text{if } j = J^{(k)} \\ \sqrt{\left(\underline{C}_{1j}^{(k)} - \overline{C}_{1(j+1)}^{(k)}\right)^2 + \left(\overline{C}_{2j}^{(k)} - \underline{C}_{2(j+1)}^{(k)}\right)^2} & \\ \quad + \sqrt{\left(\underline{C}_{1(j-1)}^{(k)} - \overline{C}_{1j}^{(k)}\right)^2 + \left(\overline{C}_{2(j-1)}^{(k)} - \underline{C}_{2j}^{(k)}\right)^2} & \text{otherwise.} \end{cases}$$

3. Let $\gamma_{max}^{(k)} = \max\{\gamma_j^{(k)} : j = 1, 2, \ldots, J^{(k)}\}$, $j^* = \text{argmax}\{\gamma_j^{(k)} : j = 1, 2, \ldots, J^{(k)}\}$, and $\Delta_{tol} > 0$ be the convergence tolerance parameter of the trust-region method.
   - If $\gamma_{max}^{(k)} \geq \delta$, $x_c^{(k)} = x_{j^*}^{(k)}$,
   - else if $\Delta_1^{(k)} \leq \Delta_{tol}$ and $\Delta_{J^{(k)}}^{(k)} \leq \Delta_{tol}$, $x_c^{(k)} = x_{j^*}^{(k)}$,
   - else if $\Delta_{J^{(k)}}^{(k)} > \Delta_{tol}$, $x_c^{(k)} = x_1^{(k)}$,
   - else $x_c^{(k)} = x_{J^{(k)}}^{(k)}$.

If $\gamma_{max}^{(k)} < \delta$, it follows that the solutions in $\mathcal{X}^{(k)}$ are all close to each other. In this case, either $x_1^{(k)}$ or $x_{J^{(k)}}^{(k)}$ is selected so that the iterate $x_c^{(k)}$ moves toward the end part of the Pareto front. In this way, the algorithm can generate well-spread solutions while searching for the minimizer of each objective as the iteration k grows.

2.2. Quadratic regression model

We construct quadratic models on a trust region centered at $x_c^{(k)}$ to search for local solutions. Suppose that $x_c^{(k)} = x_j^{(k)}$ for some $j \in \{1, 2, \ldots, J^{(k)}\}$. We set $\rho_c^{(k)} = \rho_j^{(k)}$ and $\Delta_c^{(k)} = \Delta_j^{(k)}$. The algorithm parameters η, τ, τ_{inc} and Δ_{max} are constants such that $0 \leq \eta < 1$, $0 < \tau < 1 < \tau_{inc}$ and $\Delta_{max} > 0$. Then, the trust region radius is determined by

$$\Delta^{(k)} \in \begin{cases} \left[\Delta_c^{(k)}, \min\{\tau_{inc}\Delta_c^{(k)}, \Delta_{max}\}\right] & \text{if } \rho_c^{(k)} \geq \eta \\ \{\tau\Delta_c^{(k)}\} & \text{otherwise.} \end{cases} \tag{3}$$

η is a parameter to test if the considered solution is acceptable or not. If the reduction ratio $\rho_c^{(k)}$ is greater than η, $x_c^{(k)}$ is acceptable and we either keep the same trust region or expand it. τ_{inc} is the maximum ratio to increase the trust

region radius and $\Delta_{max} > 0$ is the upper bound of it. If a solution is not acceptable, we reduce the trust region radius by factor of τ. The trust-region is defined as a closed ball centered at $x_c^{(k)}$ with radius $\Delta^{(k)}$:

$$\mathbf{B}^{(k)} = \{x \in \mathbb{R}^q \mid \|x - x_c^{(k)}\| \leq \Delta^{(k)}\}.$$

Now, we apply the quadratic least-squares regression to single objectives and build local approximate models on the trust region. Let $N^{(k)}$ be the sample size chosen according to one of the sampling schemes discussed in subsection 2.3. We sample points in $\mathbf{B}^{(k)}$ by using a central composite design [16] and evaluate the sample average objectives at the design point as well as the center point with the sample size $N^{(k)}$. Then corresponding local quadratic models are defined by

$$\hat{f}_1(x_c^{(k)} + s) \approx q_1(x_c^{(k)} + s) = c_1 + s^T \hat{g}_1 + \frac{1}{2} s^T \hat{H}_1 s,$$
$$\hat{f}_2(x_c^{(k)} + s) \approx q_2(x_c^{(k)} + s) = c_2 + s^T \hat{g}_2 + \frac{1}{2} s^T \hat{H}_2 s,$$

where $x_c^{(k)} + s \in \mathbf{B}^{(k)}$, c_1 and c_2 are scalar, and $\hat{g}_i$ and $\hat{H}_i$ are the estimates of the gradient g_i and the Hessian H_i of the quadratic model q_i at $x_c^{(k)}$ $(i = 1, 2)$.

In order to find the best trade-off solution between two conflicting objectives, we use the following form of single-objective product formulation suggested by Audet et al. [7]:

$$f_3(x) = -\prod_{i=1}^{2} \{(r_i - f_i(x))_+\}^2, \tag{4}$$

where $r_i = f_i(x_c^{(k)})$, $i = 1, 2$. The strong advantage of this formulation over the WS method is that any solution to the problem of minimizing f_3 is Pareto optimal for the original bi-objective problem, and hence solutions on both convex and nonconvex parts of the Pareto front can be detected. We also estimate f_3 using quadratic approximation:

$$\hat{f}_3(x_c^{(k)} + s) \approx q_3(x_c^{(k)} + s) = c_3 + s^T \hat{g}_3 + \frac{1}{2} s^T \hat{H}_3 s.$$

2.3. Sample size for each iteration

In the standard SAA method, the sample $N^{(k)}$ is set to be constant. However, instead of fixing a sample from the beginning and then iteratively searching for the solutions of the resulting deterministic problem (2), we may use an increasing sequence of sample sizes in order to improve the finite-time performance of the algorithm. At iteration k, we can either reuse $N^{(k-1)}$ samples from the previous iteration and draw $N^{(k)} - N^{(k-1)}$ new samples, or we can generate a new sample that is independent of all the samples generated previously. This idea of the variable sampling scheme has been studied in the context of single-objective stochastic optimization problems and shown to perform better than the standard SAA method. [13, 14].

The sample size $N^{(k)}$ is crucial to the performance of the algorithm. The difference between the solutions obtained from SAA and the solutions to the true problems can be reduced by taking a larger sample size at each iteration. If $N^{(k)}$ is too small, the resulting SAA problem would be inaccurate and may lead to a bad movement of the algorithm. However, when the radius $\Delta^{(k)}$ is still large and the solutions have not been near the Pareto front, we may not need a highly accurate approximate problem to determine solutions in the current trust region, and hence significant computational savings can be achieved by using a small size. As the number of iterations grows, the distance between the solutions from each iteration and the Pareto front decreases, and a larger number of solutions over a wide region can be evaluated, which allows the method to generate more accurate and widely spread Pareto optimal solutions. Kim and Zhang [14] show that the size of the local search region is an essential factor in the determination of the sample size. Adapting their result, we set $N^{(k)} = O\left(\ln(k)/\Delta^{(k)^2}\right)$.

2.4. Algorithm description

I. Initialization: Choose the initial point $x^{(0)}$, the initial trust-region radius Δ_0, and the value of parameters Δ_{max}, Δ_{tol}, δ η, τ, and τ_{inc}. Determine the number of design points d according to the considered central composite design. Evaluate $\hat{f}_1(x^{(0)})$ and $\hat{f}_2(x^{(0)})$ using the initial sample size $N^{(0)}$. Set $\Delta^{(0)} = \Delta_0$, and $\rho^{(0)} = 1$.

II. Algorithm: For each iteration $k = 1, 2, \ldots,$

Step 1 (next iterate selection) From the previous iteration, the set of non-dominated points $\mathcal{X}^{(k)}$ is returned. Select $x_c^{(k)}$ in $\mathcal{X}^{(k)}$ according to the procedure described in subsection 2.1.

Step 2 (trust-region radius update) Update $\Delta^{(k)}$ and $\mathbf{B}^{(k)}$ using formula (3).

Step 3 (sampling and regression models) Update $N^{(k)}$ and generate samples. Using the central composite design, select d number of design points $x_j^{(k)}, j = J^{(k)} + 1, J^{(k)} + 2, \ldots, J^{(k)} + d$. Construct quadratic regression models $q_i^{(k)}, i = 1, 2, 3$ as described in subsection 2.2.

Step 4 (solving three single-objective problems) Solve each single objective problem and let

$$x^{(k)}_{J^{(k)}+d+i} = \arg\min\{q_i^{(k)}(x) : x \in \mathbf{B}^{(k)}\},\ i = 1, 2, 3.$$

Step 5 (update the value of ρ) Evaluate $\hat{f}_i(x^{(k)}_{J^{(k)}+d+i}), i = 1, 2, 3$, and the reduction ratio

$$\rho_i^{(k)} = \frac{\hat{f}_i(x_c^{(k)}) - \hat{f}_i(x^{(k)}_{J^{(k)}+d+i})}{q_i(x_c^{(k)}) - q_i(x^{(k)}_{J^{(k)}+d+i})}. \tag{5}$$

Step 6 (update non-dominated solution set) Set

$$\begin{aligned} n_j^{(k)} &= N^{(k)}, j = J^{(k)} + 1, \ldots, J^{(k)} + d + 3, \\ \Delta_j^{(k)} &= \Delta^{(k)}, j = J^{(k)} + 1, \ldots, J^{(k)} + d + 3, \\ \rho_j^{(k)} &= \begin{cases} 1 & \text{if } j = J^{(k)} + 1, \ldots, J^{(k)} + d \\ \rho_i^{(k)} & \text{if } j = J^{(k)} + d + 1, \ldots, J^{(k)} + d + 3. \end{cases} \end{aligned}$$

Compare the vector values $\hat{F}(x)$, $x \in \{x_j^{(k)}, j = 1, 2, \ldots, J^{(k)}_{i+d+3}\}$ and determine the set of non-dominated points $\mathcal{X}^{(k+1)}$. Go to Step 1.

3. Numerical experiments

In this section, we test the proposed algorithm with several sampling schemes. The test problem is taken from [17], which is an unconstrained deterministic bi-objective problem with a convex Pareto front. After adding noise to the decision variable x, our problem is formulated as follows:

$$\begin{aligned} \text{Minimize:}\quad & f_1(x_{(1)}, x_{(2)}, \xi) = (x_{(1)}\xi_1 - 2)^2 + (x_{(2)}\xi_2 - 1)^2 \\ & f_2(x_{(1)}, x_{(2)}, \xi) = (x_{(1)}\xi_1)^2 + (x_{(2)}\xi_2 - 6)^2, \end{aligned}$$

where $\xi = [\xi_1, \xi_2] \sim [N(1, 0.1), N(1, 0.1)]$, and ξ_1 and ξ_2 are independent. We use common random numbers to generate samples from the distribution of ξ. The starting point is $x^{(0)} = [5, 5]$ and the initial parameters are set by $\Delta^{(0)} = 0.8$, $\Delta_{tol} = 0.001$, $\alpha = 0.98$, $Z^{(0)} = 5$, $\eta = 0.5$, $\tau = 0.7$, $\tau_{inc} = 1$, and $\delta = 0.001$. The Pareto front is approximated by evaluating 401×701 uniformly-spaced points on $[-2, 2] \times [0, 7]$ without the noise term ξ. We find a set of solutions $\mathcal{H}$ of around $2,500$ non-dominated solutions that are uniformly-spaced.

To evaluate our method, we use the generational distance (GD) criterion [18]. Suppose that the solution set is $\mathcal{H} = \{x_1, \ldots, x_e\}$. Then, the GD is computed by

$$GD = \frac{\sqrt{\sum_{j=1}^{e} \left\{ \min_{x_j^* \in \mathcal{H}} \|F(x_j) - F(x_j^*)\| \right\}^2}}{e}.$$

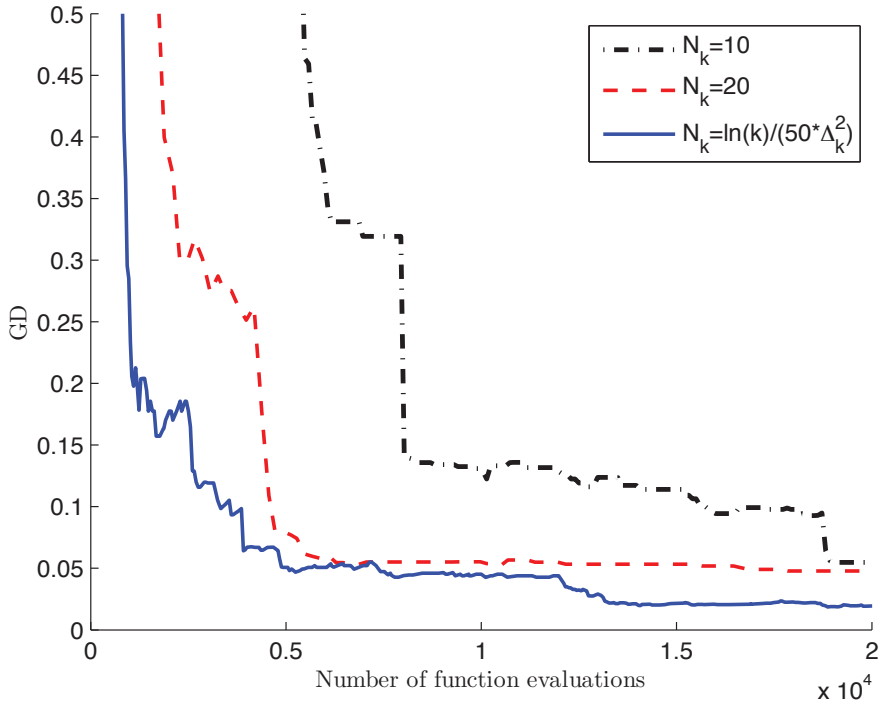


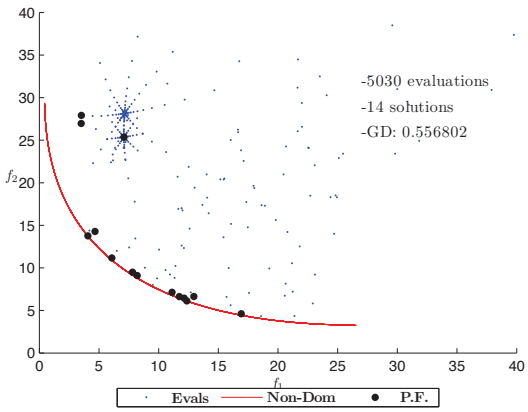
Figure 1: Comparison of different sample allocation rules

This is a measure of the average distance between the objective value at the obtained solution and the true Pareto front. Hence, smaller GD is preferable. Note that GD= 0 indicates that all the generated solutions are placed on the Pareto front.

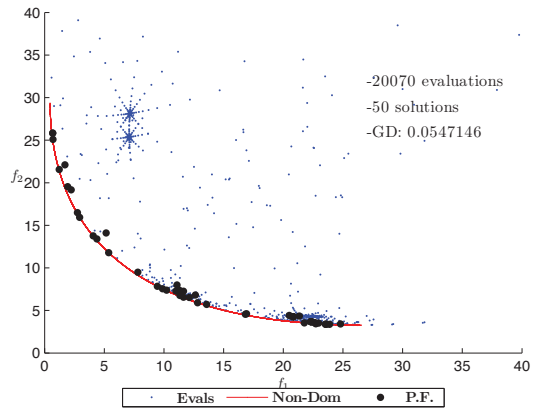
To determine the sample size at each iteration, three sampling schemes are applied: $N^{(k)} = 10$, $N^{(k)} = 20$, and $N^{(k)} = \ln(k)/(\theta * \Delta^{(k)^2})$ with $\theta = 50$. The first two schemes allocate a constant sample size to all iterations, and the third one considers the iteration number and the trust-region radius to determine the size. Figure 1 compares the values of GD as the number of function evaluations goes up to 20,000. The algorithm with $N^{(k)} = \ln(k)/(\theta * \Delta^{(k)^2})$ yields better values of GD at any number of function evaluations. In the early iterations, it takes a smaller sample size but a bigger number of iterations than the constant sampling schemes, which allows it to evaluate the objectives at more points with lower accuracy. In the later iterations, it makes a greater effort to reduce the sample variance and find more accurate solutions. In Figure 2, we compare $N^{(k)} = 10$ and $N^{(k)} = \ln(k)/(\theta * \Delta^{(k)^2})$ on the objective value space with around 5000 and 20,000 function evaluations. The small dot represents all the visited points, and the large dots are the non-dominated points generated. The solid line is the true Pareto front. Using $N^{(k)} = \ln(k)/(50\Delta^{(k)^2})$ (Figure 2(c)), almost all the generated points are near the Pareto front with around 5000 function evaluations. With $N^{(k)} = 10$, however, there are several points significantly away from the Pareto front as seen in Figure 2(a).

4. Conclusion

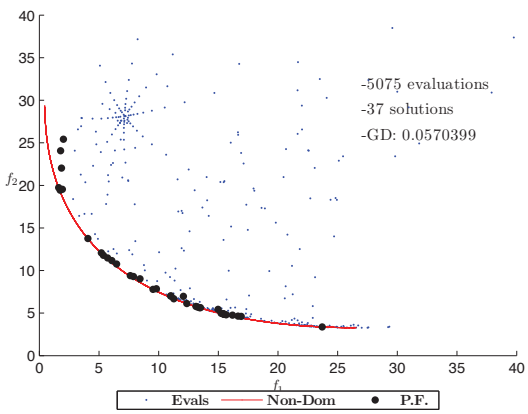
We developed an iterative algorithm for bi-objective stochastic optimization problems based on the trust region method and investigated different sampling schemes. The algorithm does not require any strong modeling assumptions, and has great potential to work well in various real-world settings. The numerical results show that the our proposed method is feasible, and the performance can be significantly improved with an appropriate sampling scheme. To improve the finite time performance of the algorithm, the sampling rule should be carefully determined with consideration for the trade-off between sampling and optimization errors. Currently, we are analyzing these two errors for



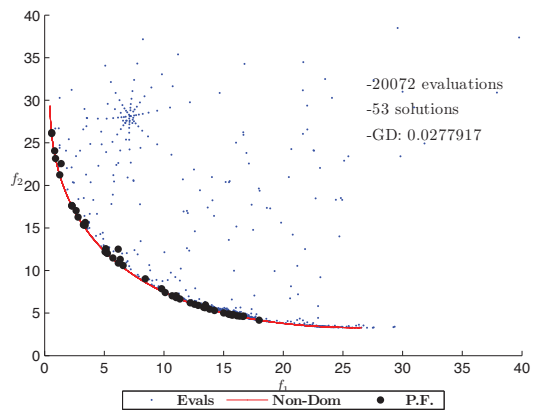
(a) Result of $N_k = 10$ with 5×10^3 evaluations



(b) Result of $N_k = 10$ with 2×10^4 evaluations



(c) Result of $N_k = \ln(k)/(50 * \Delta_k^2)$ with 5×10^3 evaluations



(d) Result of $N_k = \ln(k)/(50 * \Delta_k^2)$ with 2×10^4 evaluations

Figure 2: Performance comparisons of different sample allocation rules

a certain class of problems and developing a sampling scheme that may work robustly for those problems. To generate well-spread solutions over the Pareto front, identifying an appropriate center point of the trust region is crucial. In this regard, developing a good isolation measure is one of the key factors to enhance the performance of the algorithm. We are considering a way to directly construct the confidence interval of the distance between two objective vectors rather than use the confidence interval of each objective. Thus, the correlation between objectives can be incorporated to reduce the variance of the estimated distance.

5. Acknowledgments

This research was supported by NUS Research Grant R-266-000-049-133.

[1] B. Wilson, D. Cappelleri, T. Simpson, M. Frecker, Efficient pareto frontier exploration using surrogate approximations, Optimization and Engineering 2 (2001) 31–50.

[2] R. Tappeta, J. Renaud, J. Rodríguez, An interactive multiobjective optimization design strategy for decision based multidisciplinary design, Engineering Optimization 34 (5) (2002) 523–544.

[3] S. Shan, G. G. Wang, An efficient pareto set identification approach for multiobjective optimization on black-box functions, Journal of Mechanical Design 127 (5) (2005) 866–874.

[4] H. Markowitz, Portfolio selection: efficient diversification of investments, Blackwell, Malden, Massachusetts, 1991.

[5] R. G. Ajith Abraham, L C. Jain, Evolutionary multiobjective optimization: theoretical advances and applications, Springer-Verlag, London, 2005.
[6] J. L. Cohon, Multiobjective Programming and Planning, Academic Press, New York, 1978.
[7] C. Audet, G. Savard, W. Zghal, Multiobjective optimization through a series of single-objective formulations, SIAM Journal on Optimization 19 (2008) 188–210.
[8] J. Ryu, S. Kim, H. Wan, Pareto front approximation with adaptive weighted sum method in multiobjective simulation optimization, in: M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, R. G. Ingalls (Eds.), Proceedings of the 2009 Winter Simulation Conference, Austin, TX., 2009, pp. 623–633.
[9] K. Deb, Multi-objective genetic algorithms: Problem difficulties and construction of test problems, Evolutionary Computation 7 (1999) 205–230.
[10] L. H. Lee, E. P. Chew, S. Teng, D. Goldsman, Finding the non-dominated pareto set for multi-objective simulation models, IIE Transactions 42 (9) (2010) 656–674.
[11] A. Shapiro, Monte carlo sampling methods, in: A. Ruszczynski, A. Shapiro (Eds.), Stochastic Programming, Vol. 10 of Handbooks in Operations Research and Management Science, Elsevier, 2003, pp. 353 – 425.
[12] T. HomemDeMello, Variable-sample methods for stochastic optimization, ACM Trans. Model. Comput. Simul. 13 (2003) 108–133.
[13] G. Deng, M. C. Ferris, Variable-number sample-path optimization, Mathematical Programming 117 (2009) 81–109.
[14] S. Kim, D. Zhang, Convergence properties of direct search methods for stochastic optimization, in: B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, E. Yücesan (Eds.), Proceedings of the 2010 Winter Simulation Conference, Baltimore, MD, 2010, pp. 1003–1011.
[15] A. Conn, K. Scheinberg, L. Vicente, Introduction to derivative-free optimization, MPS-SIAM series on optimization, Society for Industrial and Applied Mathematics/Mathematical Programming Society, 2009.
[16] R. Myers, D. Montgomery, C. Anderson-Cook, Response surface methodology: process and product optimization using designed experiments, Wiley series in probability and statistics, Wiley, 2009.
[17] B. Wilson, D. Cappelleri, T. Simpson, M. Frecker, Efficient pareto frontier exploration using surrogate approximations, Optimization and Engineering 2 (2001) 31–50.
[18] D. V. Veldhuizen, G. Lamont, Multiobjective evolutionary algorithm research: A history and analysis, Tech. Rep. TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Writhe-Patterson AFB, OH (1998).